**Specification**

**Please replace the paragraph beginning on page 1, line 24 with the following:**

FIG. 11 shows a flow of steps showing the conventional procedures for the design and development of LSI (hereinafter, simply referred to as LSI design and development) in manufacture of logic circuits, systems and devices. Prior to the actual manufacturing, simulation ~~platforms~~ programs (or algorithms) are normally structured without consideration of distinctions between the hardware and software (see step B1). In step B2, verification is made as to whether the algorithms are correctly made or not. Next, isolation of the hardware and software is performed on the structured simulation ~~platforms~~ program, which are divided into hardware elements and software elements of the simulation program respectively. The isolation of the hardware and software is made by ~~experiments~~ experimentation.

**Please replace the paragraph beginning on page 2, line 8 with the following:**

In the hardware design (or H/W design), elements having equivalent functions of the algorithms are generally described by equivalent HDL source code(which stands for 'Hardware Description Language'), then, composition of circuitry is carried out (see step B3). In step B4, verification is made as to whether the sources operate ~~correct~~ correctly or not. In the software design (or S/W design), elements having equivalent functions of the algorithms are generally described by equivalent source code of a programming language having a CPU dependency (see step B5). In step B6, verification is made as to whether the sources operate correctly or not. Lastly, cooperative verification is performed on combinations of the hardware and software (see step B7).

**Please replace the paragraph beginning on page 2, line 17 with the following:**

Prior to the actual manufacturing of LSI, the procedures for the LSI design and development should meet some essential conditions ~~raising~~ regarding requirements of the system simulation and bus performance evaluation as well as the architecture design in which isolation of the hardware and software is performed by simulation. Conventionally, the system simulation is performed by the cooperative verification on the unification of the hardware and software.

**Please replace the paragraph beginning on page 3, line 22 with the following:**

It is an object of the invention to provide a bus performance evaluation method for algorithm description by which it is possible to considerably reduce turnaround times in design of LSI by excluding unwanted operations between hardware and software through ~~regarding~~ feedback loops derived from cooperative verification in the hardware design and software design. According to an aspect of the invention, these provide a method as defined in independent claim 11. Basically, this invention provides improvements in procedures for the design and development of LSI. That is, after isolation of the hardware and software being effected with respect to hardware and software sources described by the general purpose high-level language in algorithm design, an evaluation function is created to count traffic of the bus interconnecting elements to be implemented in hardware and/or in software. The sources are modified such that the evaluation function is performed every time data (e.g., a variable) is loaded onto the bus. Then, evaluation is performed on the performance of the bus having a processing rate. Based on the bus traffic that is finally produced with respect to the processing rate, isolation of the hardware and software is optimally performed at ~~the prescribed~~ a simulation stage of the architecture design. Thus, it is possible to exclude the conventional feedback loops

regarding the isolation between the hardware and software from the cooperative final verification after the actual coding. As a result, it is possible to considerably reduce the turnaround time in the design of LSI.

**Please replace the paragraph beginning on page 4, line 14 with the following:**

More specifically, the LSI design and development in manufacture is actualized by algorithm design, architecture design, actual hardware and software design, and verification. Herein, the architecture design contains a simulation platform structuring process and a bus performance evaluation process, which are interconnected by a feedback loop. In the algorithm design, sources are described by the general purpose high-level language such as the C language and C++ language. In the simulation platform structuring process, the sources are subjected to isolation of the hardware and software, while an evaluation function is created to count bus traffic of the bus interconnecting ~~between~~ the hardware and software. Every time data is written to a pre-defined variable loaded onto the bus, the evaluation function is performed to modify the sources. Then, evaluation is performed on the performance of the bus, so that the bus traffic for its processing rate is finally produced. That is, result of the bus performance evaluation process is fed back to the simulation platform structuring process such that isolation of the hardware and software is optimized in response to the bus traffic for the processing rate of the bus. This results in ~~exclusion of feedback loops derived from the cooperative verification after the actual coding~~ more efficient isolation between hardware and software elements so that hardware and software cooperative verification may be performed without constantly designating hardware and software elements, so it is possible to considerably reduce overall turnaround time of design.

**Please replace the paragraph beginning on page 6, line 22 with the following:**

Next a simulation program is structured to perform architecture design by using sources, which are used in the aforementioned algorithm design. Namely, in the simulation ~~platform~~ program structuring process, the flow proceeds to step A3 to effect isolation of the hardware and software. In step A4, an evaluation function is created. Herein, it is satisfactory that the evaluation function has an operation of counting a certain value for the bus traffic.

**Please replace the paragraphs beginning on page 7, line 3 with the following:**

In step A5, variables loaded onto the bus interconnecting ~~between~~ the hardware and software are being selected. Then, the flow proceeds to step A6 in which sources that are used in the algorithm design are modified ~~by executing~~ depending on the results of the created evaluation function when data are written to the variables loaded onto the bus, in other words, when data transfer is effected on the bus that is a subject of evaluation (hereinafter, simply referred to as the 'evaluated' bus). In response to modifications of the sources, the simulation program for use in the architecture design is structured again.

**Please replace the paragraphs beginning on page 7, line 11 with the following:**

Next, evaluation is performed on the performance of the bus, which is a subject of evaluation, by using the structured simulation program. That is, in the performance evaluation process, the flow proceeds to step A7 in which verification is performed using the structured simulation program. In step A8, bus traffic is calculated by the prescribed method, which depends on the created evaluation function. Herein, a processing rate requested by a main

function is provided from the high-level design stage. Hence, the bus traffic is calculated with respect to the processing rate of the evaluated bus in step A9.

**Please replace the paragraphs beginning on page 7, line 19 with the following:**

~~Using the bus traffic that is calculated in response to the processing rate, it~~ It is possible to check validity with respect to isolation of the hardware and software and a bus configuration. If the validity check causes a change of the bus, the sources that are described by the general purpose high-level language such as the C language and C++ language are modified, then, the simulation program is structured again and the performance evaluation is performed again. That is, the present procedures provide a feedback loop (see step A16) for feeding back the result of the performance evaluation of the bus. Due to provision of such a feedback loop, it is possible to actualize the architecture design at the high-level stage of design.

**Please replace the paragraphs beginning on page 8, line 12 with the following:**

In step A14, cooperative operational verification is performed on ~~unification of~~ the unified hardware and software. In the present flow of procedures for the design and development of LSI, the architecture design is already completed at the high-level stage of design. This places the cooperative verification as one type of simulation for making operational confirmation only, in which no design is newly made. Namely, this eliminates the necessity to provide feedback loops being derived from the cooperative verification.

**Please replace the paragraphs beginning on page 9, line 12 with the following:**

In addition, the number of times in writing the data to the variables loaded onto the evaluated bus represents a number of times in effecting data transfer on the evaluated bus, namely bus traffic. Because the processing rate requested by the main function is already known, it is possible to calculate the bus traffic for the processing rate and effect performance evaluation in accordance with the following equation (1).

(Bus traffic for the processing rate)

= (number of times in effecting data transfer) / (processing rate) ... (1)

To change the bus interconnecting ~~between~~ the hardware and software in response to the bus traffic being calculated for the processing rate, the sources used in the algorithm design are modified so that the simulation platform is to be structured again.

**Please replace the paragraph beginning on page 9, line 22 with the following:**

FIG. 3 shows an example of the 'restructured' simulation platform. With reference to the restructured simulation program shown in FIG. 3, the variable b is regarded as one that is not to be loaded onto the evaluated bus because of result of the performance evaluation of the bus. That is, the restructured simulation program has only two variables a, b that are being loaded onto the evaluated bus. Since the variable b is not loaded onto the bus, the evaluation function BUS0() is not embedded subsequent to the variable b to which data is written. In contrast, the evaluation function BUS0() is certainly embedded subsequent to the variables a, c to which data are written respectively. Thus, the evaluation function BUS0() is certainly executed just after the variables a, c to which the data are written respectively.

**Please replace the paragraph beginning on page 10, line 7, with the following:**

After restructuring of the simulation program, verification is performed by simulation. Then, the bus traffic for the processing rate is calculated again in accordance with the equation (1), so that evaluation is ~~to be~~ performed on the performance of the bus.

**Please replace the paragraph beginning on page 10, line 18, with the following:**

With reference to FIG. 4, the flow firstly proceeds to step C1 in which a specific evaluation function is created. It is satisfactory that the evaluation function meets an operation of incrementing a certain value for the bus traffic. Herein, the algorithm design uses sources that are described by the C language or C++ language. In step C2, the system reads the sources line by line while effecting syntax analysis.

**Please replace the paragraph beginning on page 15, line 23, with the following:**

Although the fifth embodiment is designed such that the variables loaded onto the evaluated bus are defined by the specific arrays, and the address transfer is effected in the main function, the fifth embodiment proceeds to calculation of the bus traffic for the processing rate and performance evaluation of the bus, then, it feeds back the result of the performance evaluation to structuring of the simulation ~~platform~~ program. Therefore, as similar to the first embodiment, the fifth embodiment allows the architecture design to be effected at the high-level stage of design.

**Please replace the paragraph beginning on page 18, line 19, with the following:**

As described heretofore, this invention has a variety of effects and technical features, which will be described below.

(1) When data are written to variables loaded onto the evaluated bus, sources that are described by the general purpose high-level language such as the C language and C++ language for use in the algorithm design are modified by executing a specific evaluation function for increment by a certain value for bus traffic, so that a simulation program is structured. Hence, bus traffic is calculated in connection with the processing rate of the bus interconnecting between the hardware and software, so bus performance evaluation can be performed at the high-level stage of design in the LSI design and development. In addition, result of performance evaluation of the bus is fed back to structuring of the simulation program, so it is possible to perform the architecture design at the high-level stage of design. Because the architecture design is performed using the sources that are described by the general purpose high-level language for use in the algorithm design, it is possible to considerably reduce overall simulation time for the architecture design. Generally speaking, as compared with the HDL, the C language and C++ language are increased in simulation speed to be approximately one-thousand times higher.

(2) Because the architecture design is performed using the sources that are described by the general purpose high-level language, it is possible to simplify feedback procedures due to the architecture design. By optimally performing isolation of the hardware and software at the prescribed stage of the architecture design, it is possible to exclude feedback loops regarding the isolation of the hardware and software from the cooperative verification after the actual coding. This guarantees considerable reduction of the turnaround time of design. As compared with the HDL and assembly languages, the C and C++ language

can be used relative small number of lines of code. That is, those languages can be easily changed and modified according to needs.

(3) This invention places the cooperative verification on unification of the hardware and software as one type of simulation for merely performing operational confirmation, wherein no design is newly performed. This brings exclusion of the feedback loops being derived from the cooperative verification. That is, it is possible to reduce a number of times in effecting simulation in RTL (Register Transfer Level), so it is possible to considerably reduce overall time and cost for the design of circuitry. Such an effect becomes substantial as the scale of circuitry being manufactured increases.